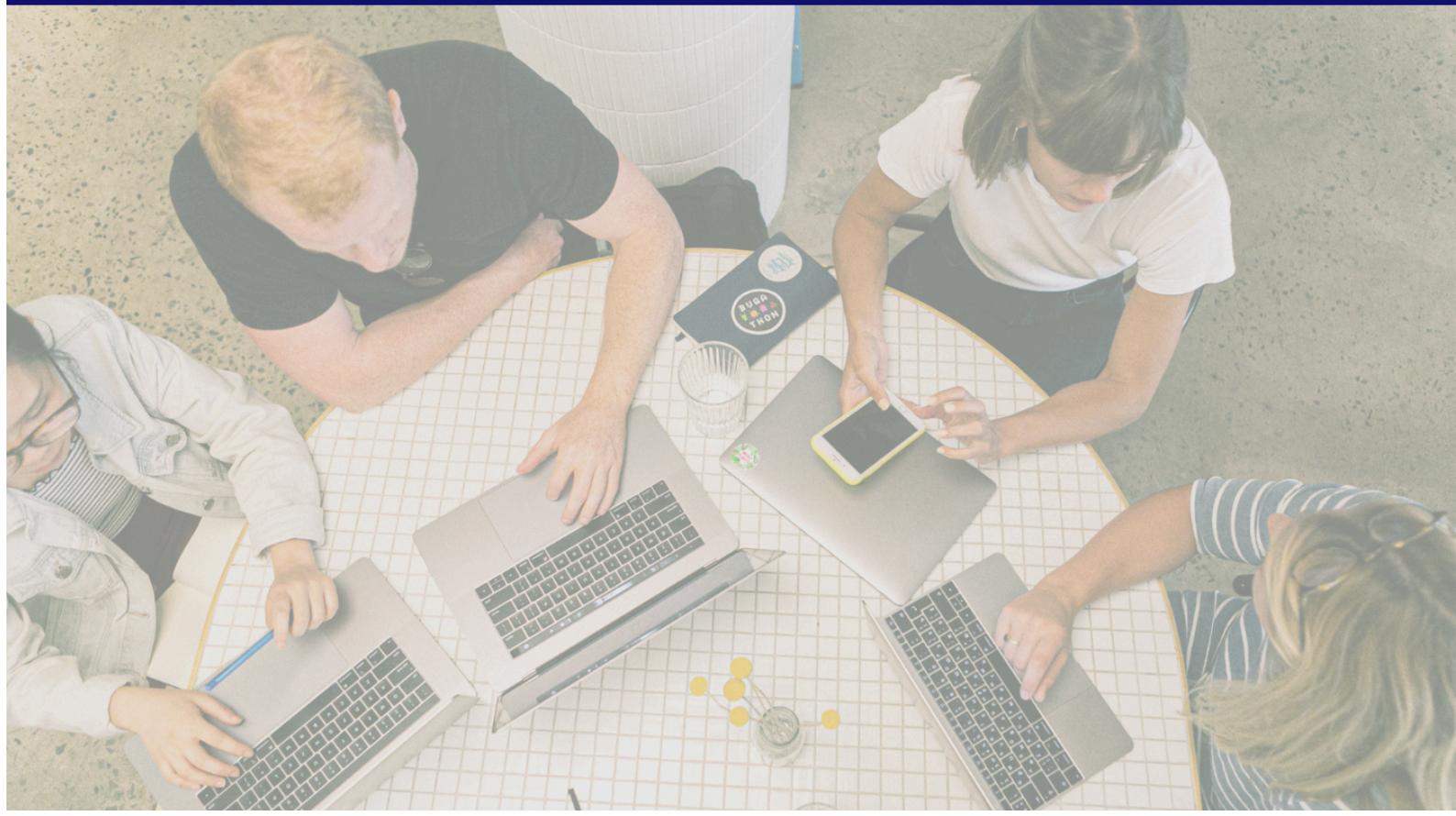


**UNifeob**  
| ESCOLA DE NEGÓCIOS



2024

# PROJETO INTEGRADO



UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
CIÊNCIA DA COMPUTAÇÃO

**PROJETO INTEGRADO**

SISTEMA DE GESTÃO E INTELIGÊNCIA DE  
NEGÓCIOS PARA ORGANIZAÇÕES SOCIAIS

**<Barbini Materiais Para Construção LTDA>**

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Marcelo Ciacco Almeida

Linguagem e Técnicas de Programação – Prof. Nivaldo de Andrade

Tópicos Avançados de Banco de Dados – Prof. Max Streicher Vallim

Computação em Nuvem – Prof. Rodrigo Marudi de Oliveira

Projeto de Computação em Nuvem – Profª. Mariângela Martimbianco Santos

Estudantes:

José Carlos Lima Junior, RA 23000394

Kaique Barbiero Bernardi, RA 23000222

Levi Vieira Junior, RA 23000444

Guilherme Carvalho, RA 23000766

SÃO JOÃO DA BOA VISTA, SP  
NOVEMBRO 2024

# SUMÁRIO

1. INTRODUÇÃO	3
2. DESCRIÇÃO DA EMPRESA	4
3. PROJETO INTEGRADO	5
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	5
3.1.1 MODELO LÓGICO	6
3.1.2 MODELO FÍSICO	7
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	8
3.2.1 APPLICATION PROGRAMMING INTERFACE ( API ) - BACK-END.	9
3.2.2 FRONT-END	11
3.3 COMPUTAÇÃO EM NUVEM	12
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	13
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	13
3.3.3 VANTAGENS DA CLOUD COMPUTING	14
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	14
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	17
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	18
3.3.7 GOOGLE CLOUD ou AWS	19
3.4 ESTRUTURA DE DADOS	21
3.4.1 LEVANTAMENTO DE REQUISITOS	21
3.4.2 VALIDAÇÃO DOS REQUISITOS	22
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	23
3.5.2 ESTUDANTES NA PRÁTICA	24
4. CONCLUSÃO	25
REFERÊNCIAS	26

# 1. INTRODUÇÃO

O Projeto Integrado (PI), intitulado “Sistema de Gestão e Inteligência de Negócios para Organizações Sociais”, tem como objetivo desenvolver e prototipar um sistema de gestão personalizado para a Empresa de Materiais de Construção Barbini Construções. Esse projeto visa oferecer uma solução que auxilia nas decisões estratégicas por meio de uma análise aprofundada dos dados financeiros e operacionais da empresa. O Sistema permitirá uma visualização clara e objetiva das informações financeiras, possibilitando uma melhor compreensão das entradas e saídas de verba, a frequência de clientes e compras, e o desempenho de produtos ao longo do tempo.

O back-end foi desenvolvido em Node.js e está hospedado na nuvem, o que facilita o gerenciamento de dados e permite integração com outros serviços em nuvem, como o armazenamento de backups automáticos e a implementação de recursos de segurança avançados. O front-end, construído com Angular, HTML e TypeScript, também está configurado para aproveitamento dos recursos da nuvem, assegurando uma experiência de usuário rápida e responsiva.

Por meio da computação em nuvem, o sistema promove a centralização das informações financeiras e operacionais, oferecendo uma visão clara e objetiva sobre entradas e saídas de recursos, frequência de clientes e compras, e desempenho dos produtos ao longo do tempo.

Essa visibilidade aprimorada facilita a tomada de decisões estratégicas, apoiando a gestão eficiente e o controle de recursos da Barbini Construções de forma contínua e sustentável.

## **2. DESCRIÇÃO DA EMPRESA**

A empresa escolhida para esse projeto é a Barbini Materiais Para Construção Ltda, mais conhecida como Barbini Construções. Situada na cidade de São Sebastião da Gramma, no interior de São Paulo, localizada na Avenida Capitão Joaquim Rabello de Andrade, nº 1061, no Centro da cidade. Registrada sob o CNPJ: 24.399.543/0001-91, a Barbini Construções oferece uma ampla variedade de materiais para construção, incluindo matéria-prima como cimento, areia e outros insumos essenciais.

Fundada em 2016 pelo dono Marcelo Barbini e por seu pai Leonildo Barbini, a empresa foi criada com a missão de promover os melhores serviços e produtos de qualidade por um preço justo. Desde sua fundação, a Barbini Construções tem se destacado no mercado local pela excelência no atendimento ao cliente e pela oferta de materiais de construção de alta qualidade, atendendo às necessidades de engenheiros, arquitetos e construtores da região.

A empresa fornece materiais de construção, como cimento, areia, brita, tijolos, blocos de concreto e outros insumos básicos. Também oferece produtos de acabamento, como pisos, azulejos, tintas e revestimentos. Além disso, presta consultoria técnica, orientando sobre a escolha dos materiais mais adequados para cada tipo de obra e dando suporte técnico para o correto uso e aplicação dos produtos.

A Barbini Construções orgulha-se de contribuir para o desenvolvimento local, oferecendo produtos de qualidade e serviços que facilitam o trabalho dos profissionais da construção civil na região de São Sebastião da Gramma.

## **3. PROJETO INTEGRADO**

### **3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS**

Para desenvolver o projeto, a equipe buscou entender a fundo o problema a ser resolvido e propor uma solução de banco de dados otimizada, com preferência pela implementação em um ambiente de computação em nuvem. A escolha de um banco de dados em nuvem proporciona escalabilidade e flexibilidade, essenciais para atender às demandas dinâmicas do sistema.

Diante dos requisitos do projeto, a equipe analisou as vantagens de utilizar um banco de dados relacional (SQL) e um banco de dados não relacional (NoSQL), considerando as características específicas de cada tipo. Bancos de dados relacionais, como o MySQL, são adequados para dados estruturados e relacionamentos complexos entre tabelas, oferecendo suporte a transações e consistência, o que é fundamental para operações financeiras e controle de estoque. Conforme explicado por Elmasri e Navathe (2015), a estrutura relacional é vantajosa em aplicações que demandam integridade referencial e um modelo bem organizado de dados.

Por outro lado, a equipe também considerou bancos de dados não relacionais, como o MongoDB, que são projetados para armazenar grandes volumes de dados não estruturados ou semiestruturados e oferecem flexibilidade na modelagem dos dados. Esse tipo de banco de dados pode ser útil em módulos que necessitam de alta velocidade em operações de leitura e escrita, como o registro de interações de clientes ou o histórico de acessos. A escolha final entre SQL e NoSQL depende da análise das necessidades específicas de cada módulo do sistema, garantindo que as características do banco de dados se alinhem aos objetivos de desempenho e escalabilidade do projeto.

Dada a natureza dos dados e a complexidade do projeto, optamos pelo uso do MySQL como solução de banco de dados relacional. O MySQL oferece a estrutura de dados e o suporte transacional necessários para manter a integridade e a consistência das informações de forma eficiente, atendendo plenamente aos requisitos do sistema.

### 3.1.1 MODELO LÓGICO

O modelo lógico do sistema foi desenvolvido para estruturar as informações de forma detalhada, utilizando como base o modelo entidade-relacionamento (ER) para criar uma representação próxima à realidade do sistema. Segundo Heuser (2009), o modelo lógico é uma etapa crucial na construção de sistemas de banco de dados, pois transforma o modelo conceitual em uma estrutura aplicável a sistemas de gerenciamento de banco de dados (SGBDs), definindo as entidades, atributos e relacionamentos de forma precisa.

Neste projeto, as entidades centrais incluem Usuário, Cliente, Vendedor, Produto, Venda, ItemVenda e Estoque.

Usuário representa os dados básicos dos usuários, formando a base para as entidades especializadas Cliente e Vendedor, as quais incluem atributos específicos conforme o tipo de usuário. Date (2012) enfatiza a importância de usar entidades especializadas para garantir que cada tipo de usuário receba o tratamento adequado em termos de armazenamento e integridade dos dados, promovendo uma modelagem mais robusta.

A entidade Produto foi estruturada para armazenar os detalhes dos itens em estoque, enquanto Estoque é responsável pelo controle das quantidades e dos limites mínimo e máximo para cada produto.

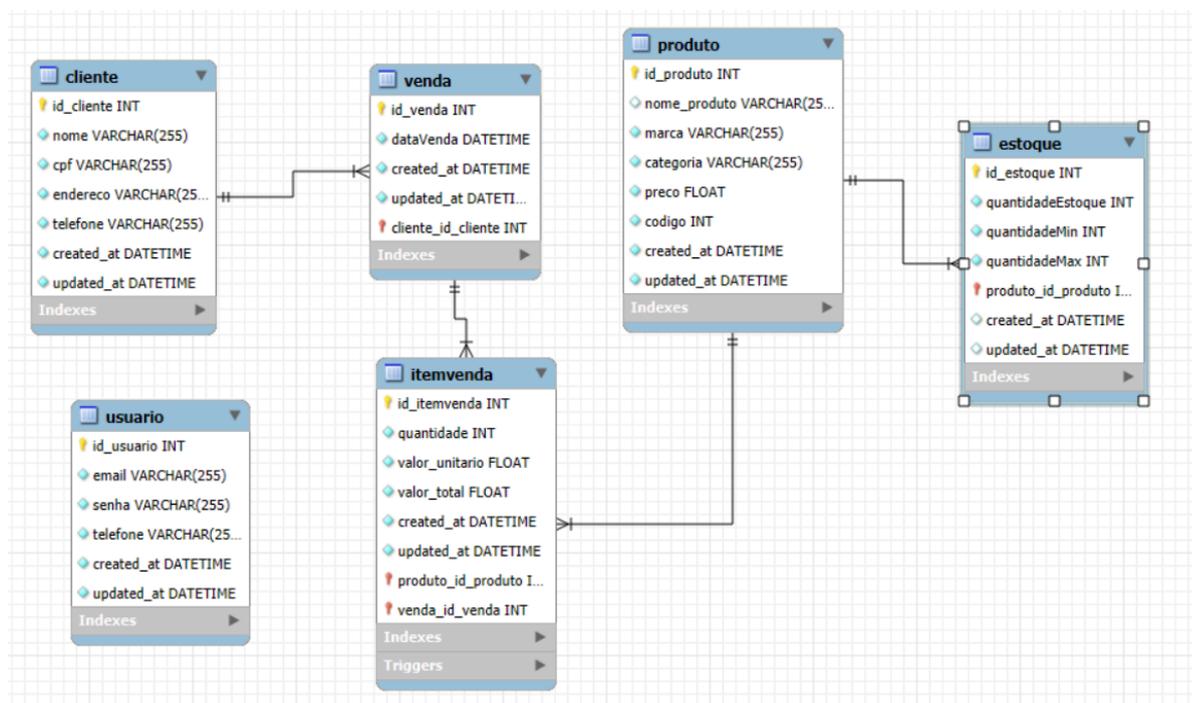
A tabela Venda registra as transações comerciais realizadas, relacionando-se com Cliente e Vendedor e armazenando o valor total da venda. De acordo com Elmasri e Navathe (2010), o uso de entidades como Venda e ItemVenda facilita o controle detalhado dos dados, permitindo uma visão granular de cada transação, especialmente em sistemas que envolvem múltiplos itens e detalhamento de produtos vendidos.

Para assegurar a integridade referencial, os relacionamentos entre as entidades foram cuidadosamente estabelecidos, como no caso de Cliente e Venda (relação 1 / N), onde um cliente pode ter várias vendas, mas uma venda pertence a um único cliente. Além disso, a entidade ItemVenda permite representar cada item de forma individual dentro de uma venda, seguindo a recomendação de orientações de modelagem de Chen (1976), que sugerem a criação de entidades intermediárias para detalhar transações e manter a flexibilidade do sistema.

Esse modelo lógico proporciona uma visão aprofundada e organizada dos dados, servindo como uma base sólida para a implementação do modelo físico. Ele detalha não só os atributos e relações entre as tabelas, mas também inclui especificações de tipo de dados, chaves primárias e estrangeiras, e atributos obrigatórios, o que garante que o sistema

mantenha integridade e funcionalidade alinhadas às exigências do projeto (Elmasri & Navathe, 2010).

**Figura 1 -Banco de Dados**



Fonte: Autores (2024)

### 3.1.2 MODELO FÍSICO

O modelo físico representa a etapa final no desenvolvimento do banco de dados para o projeto. Nesta fase, são definidos os detalhes técnicos necessários para garantir a funcionalidade e a eficiência do banco de dados, de acordo com os requisitos específicos do projeto e com o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido, o MySQL.

A construção do modelo físico envolve especificar a forma de armazenamento dos dados e criar scripts SQL para a implementação dos objetos do banco. Esses scripts incluem tabelas com definições detalhadas de tipos de dados, tamanhos e restrições de integridade para garantir a consistência e precisão dos dados. Tabelas foram projetadas para gerenciar clientes, produtos, vendas e estoque, facilitando consultas e manipulações rápidas. Conforme recomendado por Elmasri e Navathe (2015), essa etapa é essencial para traduzir o

modelo lógico em uma estrutura física implementável, o que possibilita um gerenciamento eficiente e seguro dos dados.

O modelo físico do banco de dados utiliza triggers e stored procedures como ferramentas indispensáveis para implementar a lógica de negócio diretamente no banco de dados, otimizando o desempenho e reduzindo o risco de erros humanos em operações críticas. Segundo Silberschatz et al. (2010), triggers e procedures são ferramentas essenciais para implementar a lógica de negócio diretamente no banco de dados, garantindo consistência e eficiência.

As triggers foram implementadas para automatizar a lógica de negócios associada à atualização do estoque e garantir a integridade dos dados. Por exemplo, a trigger responsável por atualizar o estoque após uma venda reduz automaticamente a quantidade do produto correspondente no estoque, garantindo que o sistema sempre reflita a quantidade atualizada dos produtos. Outra trigger foi configurada para validar o estoque antes de registrar uma venda, impedindo que a operação seja concluída caso a quantidade solicitada seja maior do que a disponível.

As stored procedures foram criadas para realizar operações frequentes, como inserções, atualizações e deleções nas principais tabelas do banco de dados, simplificando a manutenção e reutilização de código. Entre elas, destaca-se a procedure de inserção de novas vendas, que permite registrar uma venda de forma padronizada e garante que todas as colunas relevantes sejam preenchidas corretamente. Outra procedure notável é a de ajuste de estoque, que facilita a atualização das quantidades em casos de reabastecimento ou correções.

Além disso, são definidas as permissões de acesso para diferentes níveis de usuário, assegurando que apenas usuários autorizados realizem operações específicas, como inserção ou exclusão de dados sensíveis. Esse controle de acesso contribui para a segurança dos dados, assegurando que o sistema seja acessado de forma controlada e conforme os papéis de cada usuário.

Ao final, o modelo físico transforma o modelo lógico em uma estrutura de banco de dados completa, totalmente adaptada ao MySQL e integrada ao ambiente de computação em nuvem. Com essa configuração, a Barbini Construções dispõe de um sistema de banco de dados robusto e escalável, capaz de sustentar suas operações de forma segura e eficiente.

### **3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO**

No desenvolvimento do projeto, foram aplicadas técnicas avançadas de programação para garantir modularidade, reutilização de código e desempenho eficiente, baseando-se em práticas recomendadas que facilitam a manutenção e legibilidade do código. A escolha da linguagem e dos paradigmas de programação foi orientada pelas necessidades específicas do projeto, para criar uma solução robusta e sustentável.

A Programação Orientada a Objetos (POO) é essencial para organizar o código em componentes modulares, encapsulando dados e comportamentos. No sistema desenvolvido, a POO foi utilizada com a criação de classes que representam as principais entidades, como Usuário, Cliente, Produto e Venda, etc. Esse uso da POO permitiu organizar o código de forma que atributos e comportamentos específicos fossem encapsulados em cada entidade. Além disso, o uso de herança simplificou a criação de classes especializadas, como Cliente e Vendedor, promovendo a reutilização de código e reduzindo redundâncias. Essa prática resultou em uma arquitetura flexível e organizada, facilitando futuras expansões do sistema (Booch, 2007).

Para operações que demandam manipulação e processamento eficiente de dados, como a atualização de estoques e o processamento de vendas, foram aplicados princípios da programação funcional. De acordo com Clements e Northrop (2001), a programação funcional otimiza o desempenho e a clareza do código, especialmente em funções de manipulação de dados. A programação concorrente foi utilizada para operações assíncronas de acesso ao banco de dados, com o uso de funções assíncronas (`async/await`) em Node.js, promovendo o processamento simultâneo de múltiplas requisições e evitando bloqueios.

Práticas de Clean Code foram aplicadas para assegurar que o código fosse legível e sustentável. Martin (2008) destaca que um código claro e organizado facilita a manutenção e colabora para a continuidade do projeto a longo prazo. Seguindo essas diretrizes, foram adotadas boas práticas de nomenclatura descritiva, comentários relevantes e uma estrutura de pastas organizada, melhorando a clareza e a facilidade de manutenção. Além disso, foram utilizados padrões de design, como o padrão MVC (Model-View-Controller), que organiza o código em três camadas distintas. Essa organização proporciona maior modularidade e escalabilidade ao sistema, com os modelos representando dados, os controladores gerenciando a lógica de negócios e as rotas controlando as requisições.

Para o desenvolvimento da API e da interface de usuário, foram adotados frameworks e bibliotecas que aceleraram o processo e asseguraram a robustez do sistema. O uso de Node.js no back-end, conforme descrito por Tilkov e Vinoski (2010), permite a construção de APIs eficientes e escaláveis, suportadas por bibliotecas essenciais como Express para o gerenciamento de rotas e Sequelize para a manipulação de dados no banco de dados. Na interface, Angular permitiu a criação de uma aplicação SPA (Single Page Application), com componentes reutilizáveis que proporcionam uma experiência de usuário dinâmica e responsiva.

### **3.2.1 APPLICATION PROGRAMMING INTERFACE ( API ) - BACK-END.**

Para desenvolver o back-end do projeto, foi criada uma interface para suportar o gerenciamento de dados da Barbini Construções. A API foi estruturada utilizando uma arquitetura RESTful, permitindo que cada endpoint represente uma entidade ou ação específica no sistema. Essa arquitetura, conforme recomendada por Richardson e Ruby (2007), torna a API escalável e de fácil manutenção, facilitando a interação com outras aplicações e sistemas.

A equipe utilizou Node.js como base para o desenvolvimento, o que permitiu um ambiente de execução eficiente e flexível para as requisições HTTP. O Express foi adotado como o framework principal para o gerenciamento de rotas, simplificando a criação das operações CRUD (Create, Read, Update, Delete) e o processamento de requisições de forma modular. Cada endpoint foi cuidadosamente planejado para atender às operações necessárias para a gestão de clientes, produtos, vendas e estoque.

O Sequelize foi escolhido como ORM (Object-Relational Mapping), facilitando a interação com o banco de dados MySQL. O Sequelize permite que os dados sejam mapeados diretamente para modelos JavaScript, representando as tabelas no banco de dados e possibilitando operações como consultas, inserções e atualizações de maneira simplificada. Com o Sequelize, a equipe pôde definir as entidades principais, como Cliente, Produto, Venda e Estoque, e estabelecer relacionamentos entre elas, garantindo consistência nos dados e simplificando a implementação de regras de negócio.

A segurança foi uma prioridade no desenvolvimento da API, principalmente no que se refere ao acesso a dados sensíveis e operações restritas. A autenticação foi implementada utilizando JSON Web Token (JWT), uma tecnologia que assegura que apenas usuários autenticados possam acessar determinados endpoints. No momento do login, um token

único é gerado para cada usuário e é verificado em cada requisição subsequente para validar o acesso. Esse processo de autenticação com JWT, conforme indicado por Gupta e Gupta (2017), oferece uma camada robusta de segurança, fundamental para sistemas que manipulam informações sensíveis.

Além da autenticação, a API implementou uma rotina de recuperação de senha para garantir a acessibilidade do sistema mesmo em caso de esquecimento de senhas. Utilizando a integração com o serviço Twilio, a API envia mensagens de SMS para o número de telefone registrado do usuário com um código temporário ou link seguro para redefinição de senha. Esse processo envolve a verificação do número de telefone cadastrado, gerando um código único que é enviado via SMS. Essa integração com o Twilio simplifica o processo de recuperação de senha, tornando-o seguro e prático para o usuário, além de reduzir a dependência de e-mails, o que pode ser menos acessível para alguns usuários.

A organização do código foi estruturada de forma modular, com cada funcionalidade separada em controladores (controllers) específicos. Cada controller gerencia a lógica de uma entidade, como Cliente, Produto, Venda e Estoque, e é responsável por processar as requisições relacionadas a essa entidade. Esse design modular facilita a manutenção e a extensão do sistema, permitindo que novas funcionalidades sejam adicionadas com mínimo impacto sobre o restante da API. Por exemplo, o ClienteController gerencia as operações relacionadas a clientes, enquanto o ProdutoController é responsável pela manipulação dos dados de produtos, garantindo que cada parte da lógica de negócios esteja bem organizada e independente.

Para validar as funcionalidades e a integridade da API, a equipe utilizou o Postman para realizar testes detalhados em cada endpoint. O Postman permitiu simular requisições HTTP e verificar as respostas da API em tempo real, assegurando que cada operação — desde o gerenciamento de clientes e produtos até a autenticação e recuperação de senha — funcionasse conforme o esperado. Esse processo de teste garantiu que a API fosse confiável e que todos os endpoints estivessem configurados corretamente, minimizando os riscos de falhas no sistema.

### **3.2.2 FRONT-END**

No desenvolvimento do front-end do projeto, foi utilizado o framework Angular para criar uma interface dinâmica e responsiva. A aplicação foi projetada para ser acessível em diferentes dispositivos, como computadores, tablets e smartphones, garantindo uma experiência consistente aos usuários. O design responsivo permite que todas as telas, como

estoque, finanças e clientes, se adaptam automaticamente ao tamanho do dispositivo, melhorando a usabilidade.

A experiência do usuário (UX) e a interface do usuário (UI) foram aprimoradas por meio de um design intuitivo e atraente. Como recomenda Krug (2014), um design claro e de fácil navegação contribui para uma experiência fluida e sem frustrações. Por isso, a aplicação foi desenvolvida com um tema vermelho consistente, ajustando o layout e os elementos visuais para criar uma navegação clara e amigável. Foram criadas rotas para facilitar a transição entre as diferentes seções do sistema e uma barra lateral de navegação que organiza as funcionalidades principais de forma intuitiva.

No Angular, foram utilizados Reactive Forms para criar formulários robustos e dinâmicos, facilitando o registro de vendas e a adição de produtos ao estoque. Esses formulários oferecem uma validação eficiente e uma experiência de preenchimento mais fluida. Além disso, serviços como `EstoqueService` e `ClientesService` foram implementados para gerenciar dados de produtos e clientes, centralizando a lógica de acesso e manipulação de dados, o que torna a aplicação mais modular e fácil de manter.

A integração de componentes interativos, como o módulo de finanças, permite que os usuários registrem vendas e visualizem o histórico de transações de forma rápida e acessível. O uso de Angular facilita a construção de uma Single Page Application (SPA), melhorando o desempenho e a interatividade ao reduzir o tempo de carregamento das páginas.

### **3.3 COMPUTAÇÃO EM NUVEM**

A computação em nuvem é uma tecnologia fundamental para a implantação do Projeto, proporcionando a escalabilidade, segurança e flexibilidade necessárias para o crescimento e o funcionamento contínuo das operações críticas da empresa. Utilizar a nuvem como infraestrutura permite otimizar recursos, ajustar capacidade conforme a demanda e reduzir custos operacionais, ao mesmo tempo que minimiza o tempo de inatividade. Mell e Grance (2011) apontam que a computação em nuvem oferece uma maneira eficiente de acessar e gerenciar recursos computacionais remotamente, especialmente vantajosa para lidar com cargas de trabalho variáveis e demandas dinâmicas.

No contexto do projeto, a escolha de um provedor confiável como Amazon Web Services (AWS) ou Google Cloud Platform (GCP) permite integrar tecnologias essenciais, incluindo armazenamento em nuvem, máquinas virtuais e balanceamento de carga. Essas

soluções facilitam o funcionamento ininterrupto e escalável do sistema, alinhando-se com as necessidades operacionais da empresa e melhorando a eficiência no uso de grandes volumes de dados (Armbrust et al., 2010).

Além dos benefícios de desempenho, a computação em nuvem promove os princípios de TI Verde ao reduzir a necessidade de infraestrutura física e, conseqüentemente, o consumo de energia e a emissão de carbono. A nuvem permite que a Barbini Construções contribua para a sustentabilidade, já que os data centers dos principais provedores de nuvem são otimizados para eficiência energética, com alguns até operando com energia renovável. Segundo Murugesan (2008), o conceito de TI Verde engloba práticas e tecnologias que minimizem o impacto ambiental, sendo a computação em nuvem uma ferramenta central para reduzir o consumo de recursos ao mover processos para infraestruturas centralizadas e sustentáveis.

A implementação em nuvem também permite incorporar práticas avançadas de segurança e monitoramento, essenciais para proteger dados sensíveis e manter a integridade das operações. Com backups automáticos e mecanismos de recuperação de desastres, o sistema da Barbini se torna resiliente e preparado para eventuais falhas, o que é crucial para a continuidade do serviço e o suporte ao crescimento da empresa.

### **3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING**

O projeto de computação em nuvem para a Barbini Construções busca aprimorar a eficiência operacional, reduzir custos e aumentar a escalabilidade, permitindo à empresa uma gestão mais integrada e ágil. A nuvem centraliza as operações, oferecendo acesso em tempo real aos dados de estoque, vendas e clientes, o que facilita a tomada de decisão e elimina atualizações manuais. Além disso, a computação em nuvem reduz custos operacionais ao dispensar servidores físicos e permite um modelo de pagamento conforme o uso, adequado às necessidades variáveis da empresa (Armbrust et al., 2010).

Outro objetivo essencial é a escalabilidade, que permite ajustar a capacidade de processamento e armazenamento conforme a demanda, mantendo a operação eficiente em qualquer cenário. A nuvem também oferece alta disponibilidade e segurança, com backups automáticos e recuperação de desastres, garantindo a continuidade das operações e a proteção dos dados.

### **3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO**

A computação em nuvem é aplicada no projeto da Barbini Construções para otimizar a gestão de operações e dados de maneira eficiente e flexível. Com a nuvem, a empresa pode centralizar dados de estoque, vendas e clientes, acessando essas informações em tempo real e promovendo uma visão integrada das operações. Esse acesso rápido facilita a coordenação entre departamentos e melhora a tomada de decisão. Por exemplo, a equipe de vendas consegue verificar o estoque em tempo real antes de fechar pedidos, enquanto o setor de compras pode programar reposições com base nos níveis mínimos e máximos definidos no sistema.

Um dos principais benefícios da computação em nuvem é a flexibilidade para ajustar a infraestrutura de acordo com as demandas do negócio. Isso permite à Barbini Construções aumentar ou reduzir o uso de recursos conforme a necessidade, garantindo que o sistema funcione de maneira eficiente sem custos fixos elevados. Além disso, a agilidade no desenvolvimento de soluções é um fator essencial. A nuvem permite implementar novas funcionalidades ou realizar atualizações no sistema sem a complexidade de gerenciar uma infraestrutura local, o que acelera o processo de inovação (Buyya et al., 2013).

A computação em nuvem também traz uma significativa redução de custos operacionais. Ao utilizar uma infraestrutura em nuvem, a Barbini Construções evita investimentos iniciais elevados em servidores e equipamentos, substituindo-os por um modelo de pagamento conforme o uso. Esse modelo permite alocar recursos financeiros de maneira mais estratégica, aplicando-os diretamente em atividades essenciais para o crescimento da empresa.

### **3.3.3 VANTAGENS DA CLOUD COMPUTING**

A computação em nuvem oferece diversas vantagens estratégicas para a Barbini Construções, especialmente considerando sua necessidade de crescimento e controle de custos. Uma das principais vantagens é a elasticidade dos recursos, que permite à empresa expandir ou reduzir a capacidade de processamento e armazenamento conforme a demanda. Isso é especialmente útil durante picos de vendas ou sazonalidades, garantindo que o sistema se adapte sem necessidade de infraestrutura física adicional.

O modelo de pagamento conforme o uso é outro benefício relevante, pois elimina altos custos iniciais e permite que a Barbini pague apenas pelos recursos realmente

utilizados. Essa flexibilidade financeira otimiza o uso dos recursos da empresa e libera capital para ser investido em áreas estratégicas. Além disso, a alta disponibilidade oferecida pela nuvem assegura que o sistema esteja acessível 24/7, com mecanismos de backup e recuperação de desastres integrados, reduzindo riscos de interrupção.

A segurança aprimorada é outro ponto essencial, já que a infraestrutura em nuvem fornece ferramentas avançadas de proteção de dados, como criptografia e controles de acesso rigorosos, garantindo a confidencialidade das informações da Barbini Construções. Em termos financeiros, a nuvem oferece uma redução significativa nos custos operacionais e de capital, substituindo servidores locais por uma infraestrutura escalável e eficiente.

### **3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING**

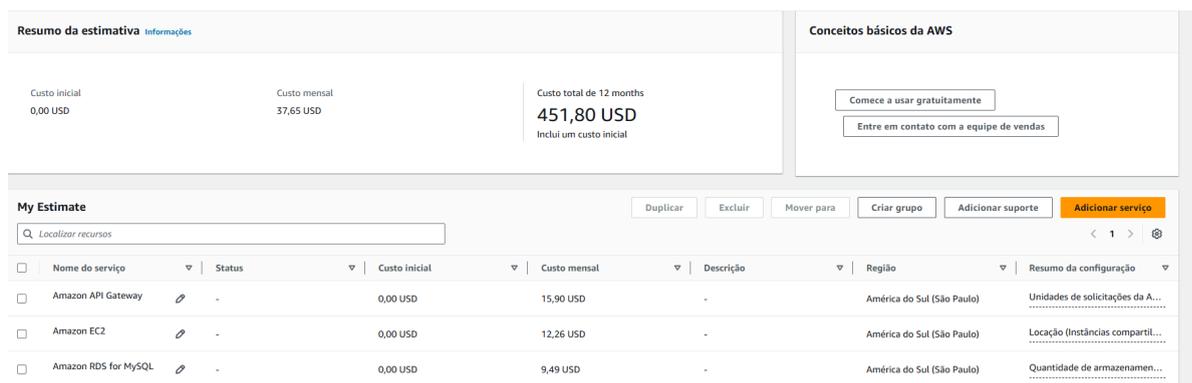
No projeto da Barbini Construções, a implantação na nuvem adota uma combinação de Infrastructure as a Service (IaaS) e Platform as a Service (PaaS) para maximizar a eficiência e a flexibilidade do sistema. O uso de IaaS oferece à empresa controle completo sobre servidores virtuais, armazenamento e redes, assegurando uma execução segura e escalável das aplicações. Com o PaaS, a equipe de desenvolvimento pode focar no código e nas funcionalidades do sistema, sem se preocupar com a gestão da infraestrutura subjacente, acelerando o desenvolvimento das soluções e garantindo agilidade nas atualizações.

Adicionalmente, a integração com ferramentas de Software as a Service (SaaS) permite o uso de soluções prontas e acessíveis via navegador, como ferramentas de colaboração e produtividade, que apoiam a comunicação interna, sem a necessidade de instalações locais.

Para a implantação do sistema do Projeto, foi realizado um estudo comparativo entre Amazon Web Services (AWS), Google Cloud Platform (GCP) e Microsoft Azure. Abaixo estão as configurações básicas e os custos estimados para cada provedor, utilizando os serviços mais adequados para atender aos requisitos do sistema.

Na configuração da AWS conforme a imagem abaixo, temos o uso de IaaS para instâncias de máquinas virtuais via Amazon EC2, o banco de dados relacional Amazon RDS com MySQL e o armazenamento de arquivos por meio do Amazon S3. Um balanceador de carga (ELB) é configurado para garantir a estabilidade e a disponibilidade. O custo mensal estimado é de USD 37,65, com Amazon API Gateway (USD 15,90), Amazon EC2 (USD 12,26) e Amazon RDS for MySQL (USD 9,49). A AWS oferece uma combinação de flexibilidade e segurança, ideal para empresas que precisam de personalização em grande escala.

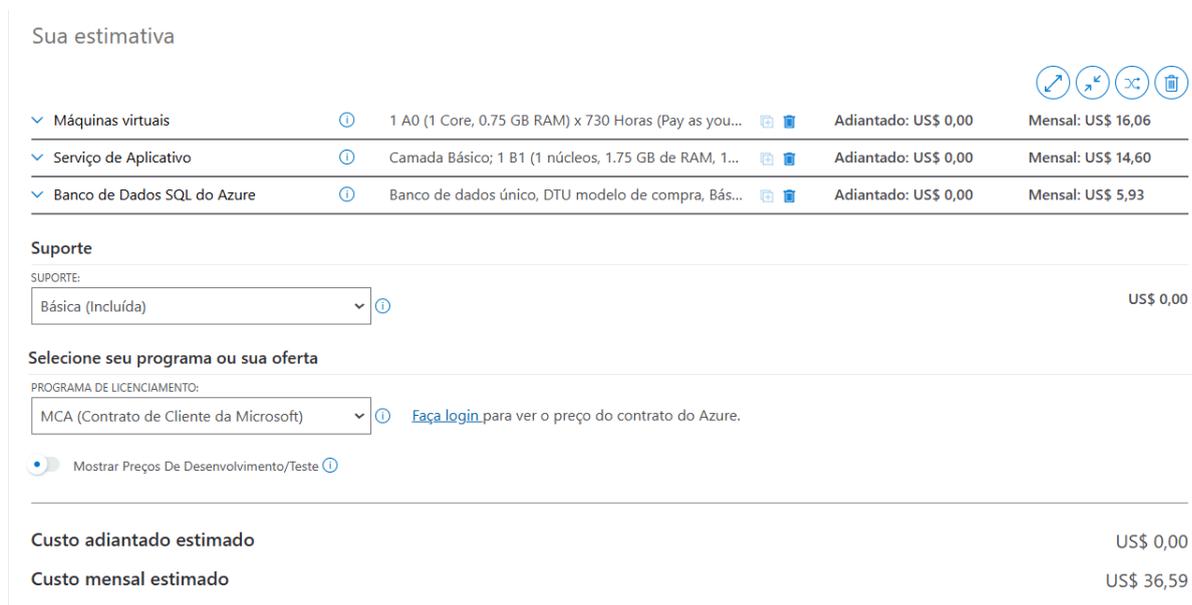
## Figura 2 - Orçamento Amazon Web Services (AWS)



Fonte: Autores (2024)

No Azure, a configuração inclui uma máquina virtual básica, o Serviço de Aplicativo para hospedagem e um banco de dados SQL gerenciado pelo Azure SQL Database. O custo mensal estimado é de USD 36,59, com Máquinas Virtuais (USD 16,06), Serviço de Aplicativo (USD 14,60) e Banco de Dados SQL do Azure (USD 5,93). O Azure é bem integrado com o ecossistema Microsoft, oferecendo uma boa opção para empresas que já utilizam as soluções Microsoft, com forte custo-benefício e suporte para desenvolvimento de aplicativos.

## Figura 3 - Orçamento Microsoft Azure

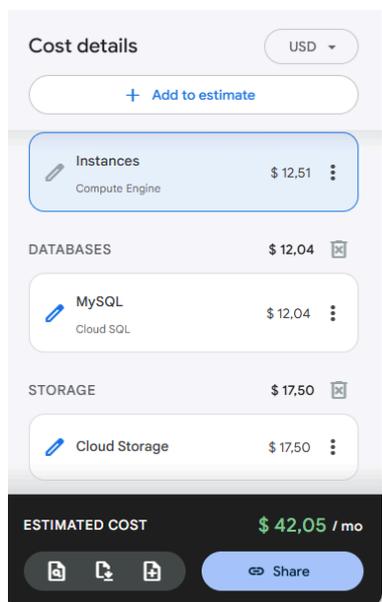


Fonte: Autores (2024)

No Google Cloud, a configuração inclui o Compute Engine para instâncias de máquinas virtuais, Cloud SQL com MySQL para o banco de dados e Cloud Storage para armazenamento de arquivos. O custo mensal estimado é de USD 42,05, com Compute Engine (USD 12,51), Cloud SQL (MySQL) (USD 12,04) e Cloud Storage (USD 17,50). O

GCP se destaca pelo armazenamento escalável e pela integração com ferramentas de análise e machine learning, sendo ideal para empresas com foco em dados.

**Figura 3 - Orçamento Google Cloud Platform (GCP)**



Fonte: Autores (2024)

Comparando o custo total mensal, o Azure se apresenta como a opção mais econômica (USD 36,59), seguido pela AWS (USD 37,65), e o Google Cloud apresenta um custo ligeiramente mais elevado (USD 42,05). Em termos de vantagens, a AWS oferece uma ampla gama de serviços e personalização de infraestrutura, ideal para configurações flexíveis. O Azure integra-se bem com o ecossistema Microsoft, sendo econômico e bem suportado para desenvolvimento e hospedagem de aplicações. O Google Cloud destaca-se em armazenamento e análise de dados, ideal para empresas com foco em escalabilidade e integração com ferramentas de dados.

Considerando o custo e os requisitos do projeto da Barbini Construções, tanto a AWS quanto o Azure oferecem uma estrutura econômica e com as funcionalidades necessárias para suportar o sistema. No entanto, a AWS se destaca pela flexibilidade e escalabilidade, enquanto o Azure proporciona uma integração robusta com soluções da Microsoft e é mais econômico. O Google Cloud, por outro lado, é ideal para empresas que priorizam armazenamento escalável e ferramentas de análise de dados.

### **3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)**

Para atender aos objetivos e requisitos da Barbini Construções, foi realizada uma análise entre Google Cloud e Amazon Web Services (AWS) como potenciais provedores de nuvem. Com base nos critérios de escalabilidade, flexibilidade, segurança e custo, a AWS foi selecionada por sua gama abrangente de serviços e pela infraestrutura robusta, que se alinha aos objetivos de desempenho e confiabilidade do projeto.

A AWS se destaca por sua infraestrutura global consolidada e ampla oferta de serviços que facilitam o crescimento e a eficiência da Barbini Construções. Para o processamento e hospedagem da API, o Amazon EC2 foi escolhido devido à sua capacidade de ajuste dinâmico e seu modelo de pagamento conforme o uso, permitindo que a empresa escale recursos de computação conforme a demanda (Amazon Web Services, 2023). A instância t2.micro atende ao projeto com uma solução de baixo custo e confiável, enquanto a escolha da região América do Sul (São Paulo) visa minimizar a latência para os usuários locais.

Para a gestão de banco de dados, o Amazon RDS com MySQL foi selecionado por oferecer um serviço de banco de dados gerenciado e seguro, com capacidade de escalabilidade horizontal e vertical. A instância db.t4g.medium combina desempenho e custo-benefício, permitindo à Barbini gerenciar e escalar suas operações de dados sem comprometer a performance.

Além disso, o Amazon SNS foi adotado para envio de notificações em massa aos usuários, devido à sua escalabilidade e facilidade de integração, tornando-o ideal para o volume projetado de mensagens (Brock, 2022). O Amazon S3, por sua vez, fornece uma solução de armazenamento seguro e escalável para os arquivos estáticos do front-end Angular, com armazenamento flexível e expansão automática.

A AWS oferece um conjunto robusto de recursos de segurança, incluindo o AWS Identity and Access Management (IAM) para controle granular de acesso, criptografia de dados em repouso e em trânsito, e conformidade com padrões internacionais como ISO 27001 e SOC 2/3 (Murphy, 2021). Esses recursos asseguram que a Barbini Construções mantenha a conformidade com as melhores práticas do setor, garantindo a proteção dos dados e a continuidade das operações, mesmo em caso de falhas.

Com base nos critérios de custo, segurança, escalabilidade e suporte, a AWS foi escolhida como a opção ideal para o projeto da Barbini Construções. Seus serviços integrados, infraestrutura global e compromisso com segurança e conformidade fornecem

um ambiente de nuvem flexível, econômico e seguro para apoiar o crescimento e a eficiência operacional da empresa.

### **3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING**

O projeto da Barbini Construções utiliza uma combinação de modelos de cloud computing para maximizar eficiência e flexibilidade, adotando Infrastructure as a Service (IaaS) e Platform as a Service (PaaS). A escolha do IaaS permite que a empresa tenha controle sobre servidores virtuais, armazenamento e redes, oferecendo uma infraestrutura personalizável e escalável, enquanto o PaaS simplifica o desenvolvimento e gerenciamento de aplicações ao fornecer um ambiente pronto para deploy. Dessa forma, o projeto pode expandir sua capacidade conforme a demanda, utilizando recursos conforme o necessário.

O balanceamento de carga desempenha um papel crucial na arquitetura de cloud computing da Barbini Construções, pois distribui automaticamente as requisições entre diferentes instâncias de servidor. Esse mecanismo não apenas otimiza o uso dos recursos, mas também aumenta a disponibilidade e a resiliência do sistema, reduzindo a possibilidade de falhas e garantindo que os usuários experimentem tempos de resposta consistentes. Durante períodos de pico, o balanceamento de carga é especialmente importante, evitando a sobrecarga de servidores individuais e mantendo a estabilidade da aplicação.

A anatomia de cloud computing envolve componentes essenciais, como servidores virtuais, bancos de dados gerenciados e redes de armazenamento. Esses elementos trabalham em sinergia para criar uma infraestrutura robusta e escalável, permitindo que diferentes serviços se integrem e operem com eficiência. Em conjunto com sistemas de monitoramento em tempo real, essa infraestrutura permite ajustes automáticos de capacidade e detecta problemas antes que eles impactem os usuários, oferecendo uma experiência contínua e confiável.

Entre os paradigmas tecnológicos subjacentes, destacam-se a escalabilidade elástica, que permite o ajuste dinâmico de recursos de acordo com a demanda, e o pagamento por uso, um modelo que otimiza custos ao permitir que a empresa pague apenas pelos recursos efetivamente utilizados. O projeto também se beneficia de virtualização, que permite o isolamento dos ambientes de desenvolvimento e produção, proporcionando flexibilidade e segurança. Esses paradigmas juntos oferecem uma plataforma eficiente e econômica, alinhada às necessidades dinâmicas de crescimento e inovação da Barbini Construções.

### **3.3.7 GOOGLE CLOUD ou AWS**

Para implantar o projeto na nuvem, foi criado um plano de ação detalhado utilizando a Amazon Web Services (AWS) como plataforma. Esse plano abrange configurações essenciais para garantir a segurança, escalabilidade e monitoramento do sistema, assegurando que a infraestrutura atenda aos requisitos do projeto de maneira eficiente e econômica.

O primeiro passo é a configuração das instâncias EC2, que hospedarão a API da aplicação. Serão criadas instâncias no Amazon EC2 com sistema operacional Linux, visando a redução de custos. As especificações de CPU, memória e armazenamento serão ajustadas conforme as necessidades do sistema, garantindo que os recursos estejam adequados à demanda. Além disso, o firewall (Security Groups) será configurado para gerenciar o tráfego e proteger o acesso à aplicação, criando um ambiente seguro para as operações.

Para o banco de dados, o sistema utilizará o Amazon RDS com MySQL como solução gerenciada. A instância db.t2.micro foi escolhida para se manter dentro do nível gratuito, com 20 GB de armazenamento SSD como base inicial. Backups automáticos e réplicas serão configurados, garantindo a recuperação dos dados.

O Elastic Load Balancer (ELB) será implementado para distribuir o tráfego entre as instâncias EC2, garantindo estabilidade e performance. O ELB também permite a detecção de anomalias e failover automático, assegurando a continuidade do serviço durante picos de acesso e protegendo a confiabilidade da aplicação.

Para o armazenamento de arquivos estáticos, como imagens e documentos, será utilizado o Amazon S3. Será configurado um bucket com 5 GB de armazenamento inicial, suficiente para as necessidades do projeto. As políticas de acesso ao bucket serão ajustadas para garantir que apenas os arquivos necessários estejam disponíveis publicamente, protegendo os dados sensíveis dos usuários.

O AWS Identity and Access Management (IAM) será configurado para definir permissões e controle de acesso, garantindo que cada usuário tenha acesso apenas às áreas necessárias. A autenticação multifator (MFA) será habilitada para administradores e outros usuários com privilégios elevados, reforçando a segurança. Além disso, o AWS Shield será ativado para proteger o sistema contra ataques DDoS, assegurando que o sistema continue seguro e disponível.

O AWS CloudWatch será utilizado para monitorar o desempenho e o uso dos recursos, como instâncias EC2, RDS e S3. Alertas serão configurados para acompanhar o uso de CPU e memória, permitindo ajustes rápidos para maximizar a eficiência e otimizar os custos. A seção de Custos e Cobrança do console AWS permitirá monitorar o consumo, e, se

o uso se aproximar do limite de orçamento de US\$ 100, ajustes serão realizados para evitar gastos desnecessários.

Para cenários de implantação, o sistema pode operar com configurações básicas durante períodos de carga leve. Em momentos de alta demanda, o ELB e o CloudWatch ajustam automaticamente os recursos, permitindo que o sistema se adapte sem comprometer a eficiência e o custo. Esse plano de ação assegura que o sistema aproveite as vantagens da AWS, oferecendo uma solução escalável, segura e econômica.

### **3.4 ESTRUTURA DE DADOS**

No projeto, foi realizado um levantamento de requisitos que orienta a escolha e aplicação das estruturas de dados no projeto. Esse levantamento é essencial para garantir que as operações de armazenamento, consulta e manipulação de dados sejam eficientes e escaláveis, em consonância com a linguagem de programação e o ambiente de computação em nuvem adotados. De acordo com Cormen et al. (2009), a escolha de estruturas de dados apropriadas é fundamental para otimizar o desempenho do sistema, especialmente em operações de grande volume e alta frequência, como aquelas que serão realizadas no projeto.

A seleção das estruturas de dados foi guiada pelas necessidades de cada módulo do sistema, considerando operações específicas, como inserção, busca e atualização. Estruturas como listas, tabelas de dispersão e filas foram escolhidas com base em sua capacidade de manipular grandes volumes de dados de forma rápida e precisa, conforme recomendado por Weiss (2012), que destaca a importância da adequação algorítmica e estrutural ao contexto de uso.

#### **3.4.1 LEVANTAMENTO DE REQUISITOS**

No levantamento de requisitos para o projeto, foi realizada uma análise cuidadosa para definir as estruturas de dados ideais para cada módulo, levando em conta a natureza dos dados, a frequência das operações e os requisitos de desempenho. De acordo com Cormen et al. (2009), a escolha da estrutura de dados influencia diretamente a eficiência do sistema, e uma seleção cuidadosa é essencial para otimizar o tempo de resposta e o uso de memória, particularmente em sistemas com operações intensivas como o da Barbini.

Para o módulo de gerenciamento de clientes, foi selecionada uma estrutura de **listas**. As listas são eficientes para operações que envolvem adição e atualização de registros em grande quantidade, proporcionando uma estrutura que facilita o acesso direto aos dados dos clientes. Essa escolha garante que o sistema seja responsivo às frequentes operações de busca e atualização realizadas por usuários em tempo real.

No módulo de controle de estoque, onde a agilidade nas consultas e o controle rápido dos níveis de inventário são essenciais, foi adotada uma tabela de dispersão (hash table). Segundo Weiss (2012), tabelas de dispersão são ideais para operações de busca frequente e de atualização rápida, reduzindo significativamente o tempo de resposta. Ao associar produtos a chaves únicas, a hash table garante eficiência no acesso direto e facilita o gerenciamento de estoque da Barbini.

Para o módulo de finanças, que precisa registrar e ordenar transações sequencialmente, foi implementada uma estrutura de filas. Filas são apropriadas para cenários em que a ordem de inserção deve ser mantida, como o processamento de vendas em sequência. Esse modelo permite que o sistema gerencie o fluxo financeiro em ordem cronológica, facilitando o acompanhamento de cada transação conforme ela ocorre.

Além disso, foram considerados os requisitos de desempenho em cada módulo para garantir uma estrutura que minimize o uso de memória e maximize a eficiência, especialmente em operações frequentes de busca e atualização. Conforme recomendado por Cormen et al. (2009), a análise da complexidade algorítmica orientou a escolha de cada estrutura de dados, garantindo uma aplicação eficiente e adequada às necessidades específicas de cada funcionalidade.

### **3.4.2 VALIDAÇÃO DOS REQUISITOS**

A validação dos requisitos, foi realizada uma análise detalhada da integração das estruturas de dados com o ambiente de computação em nuvem, visando otimizar o armazenamento e o gerenciamento de grandes volumes de dados de forma distribuída. A escolha das estruturas de dados considerou a escalabilidade e a capacidade de lidar com o aumento de dados, garantindo que o sistema mantenha seu desempenho, mesmo sob cargas variáveis.

Em um ambiente de nuvem, a elasticidade e a capacidade de distribuir cargas são fundamentais. A implementação das estruturas de dados, como tabelas de dispersão para controle de estoque e listas para gerenciar registros de clientes, foi feita levando em conta a

alta demanda por operações de leitura e escrita, garantindo que o sistema possa escalar automaticamente em resposta ao aumento de uso. Conforme observado por Ullman e Widom (2008), a distribuição de dados em uma arquitetura de nuvem exige estruturas que suportem o acesso rápido e simultâneo, de modo a otimizar a resposta do sistema e reduzir a latência.

Além disso, a validação assegura que as estruturas de dados ofereçam integridade e disponibilidade. As tabelas de dispersão, por exemplo, otimizam o acesso direto e o controle de inventário, minimizando o tempo de busca, enquanto as filas e listas, utilizadas para transações e clientes, proporcionam uma organização que garante consistência de dados em tempo real. Em alinhamento com os princípios de elasticidade da computação em nuvem, essas estruturas foram testadas para garantir que se adaptam dinamicamente ao volume de operações, sem comprometer a integridade dos dados.

Por fim, o sistema foi configurado para aproveitar a distribuição de carga da infraestrutura de nuvem, garantindo que cada operação seja realizada de forma eficiente e que os dados estejam sempre acessíveis aos usuários. Esse processo de validação é essencial para que o sistema atenda aos requisitos de integridade, eficiência e alta disponibilidade, fundamentais para o sucesso do em um ambiente em nuvem.

### **3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS**

#### **1. Estereótipo e Convívio Social:**

Estereótipos são generalizações simplificadas sobre grupos de pessoas, que influenciam como nos relacionamos socialmente. Eles podem moldar nossas percepções e interações com os outros, muitas vezes levando a preconceitos e discriminação.

**Exemplo prático:** Imagine uma pessoa que acredita que todos os adolescentes são irresponsáveis e desinteressados na escola. Esse estereótipo pode afetar como ela interage com jovens, talvez esperando menos deles ou tratando-os com desdém. Isso pode prejudicar o relacionamento e criar barreiras na comunicação e na compreensão mútua.

#### **2. Estereótipo e Representação:**

Estereótipos também influenciam a forma como as pessoas são representadas na mídia e na cultura. A forma como grupos são retratados pode reforçar ou desafiar esses estereótipos.

**Exemplo prático:** Em filmes e séries, personagens de diferentes origens étnicas podem ser retratados de maneiras estereotipadas, como os latino-americanos sendo frequentemente mostrados como trabalhadores braçais. Esse tipo de representação pode perpetuar uma visão limitada e negativa desses grupos, impactando a percepção pública e o tratamento que recebem na vida real.

### **3. Troco Likes: A Idealização da Vida na Internet:**

A busca por aprovação nas redes sociais leva à idealização da vida. As pessoas frequentemente compartilham apenas os aspectos positivos e curados de suas vidas, criando uma imagem muitas vezes irrealista que pode impactar a autoestima e a percepção dos outros.

**Exemplo prático:** Alguém que vê fotos de viagens luxuosas e eventos sociais extravagantes de amigos na internet pode sentir que sua própria vida é menos interessante ou bem-sucedida. Esse fenômeno pode levar a sentimentos de inadequação e ansiedade, já que a comparação constante com essas representações idealizadas pode distorcer a realidade e afetar a saúde mental.

### **4. Convivendo com a Diferença:**

Convivendo com a diferença envolve a aceitação e o respeito por pessoas que possuem características, crenças e culturas diferentes das nossas. Essa convivência pode enriquecer nossas experiências e promover uma sociedade mais inclusiva e harmoniosa.

**Exemplo prático:** Em um ambiente de trabalho diversificado, conviver com colegas de diferentes origens culturais e perspectivas pode levar a soluções mais criativas e inovadoras para problemas. A colaboração entre pessoas com diferentes experiências de vida pode resultar em um ambiente mais enriquecedor e produtivo.

## **3.5.2 ESTUDANTES NA PRÁTICA**

Após realizarmos um estudo sobre o tema “Estereótipo das pessoas do interior” foi desenvolvido um podcast sobre com o tema “Desconstruindo Estereótipos” em que os membros do grupo comentam sobre esse assunto.

link do podcast: <https://youtu.be/h3MfujxGaZk>

## 4. CONCLUSÃO

O desenvolvimento do projeto representou uma importante oportunidade para aplicar conhecimentos em computação em nuvem, banco de dados e desenvolvimento de aplicações web. Ao longo do Projeto Integrado (PI), foram abordados tópicos centrais, como a escolha de uma infraestrutura de nuvem escalável e sustentável, o design de um banco de dados relacional eficiente com MySQL, e o uso do framework Angular para criar um front-end dinâmico e responsivo.

Uma das maiores dificuldades enfrentadas foi a integração das diversas tecnologias, garantindo que todas as partes — back-end, banco de dados e front-end — funcionassem de maneira harmoniosa e segura. A implementação de práticas de segurança robustas, especialmente em um ambiente de nuvem, exigiu uma atenção especial, visto que o sistema precisava garantir a proteção de dados sensíveis. A definição das estruturas de dados mais adequadas para cada módulo, visando eficiência e escalabilidade, também foi um desafio que exigiu análise cuidadosa.

O uso da AWS como plataforma de nuvem foi decisivo para que o sistema se beneficiasse de recursos avançados de monitoramento, segurança e flexibilidade. Com a configuração de instâncias EC2, RDS para o banco de dados MySQL, balanceamento de carga com ELB e armazenamento seguro no S3, a Barbini Construções obteve uma infraestrutura eficiente que atende às suas necessidades operacionais, além de garantir que o sistema seja adaptável ao crescimento futuro.

O conceito de TI Verde, incorporado no projeto, demonstrou a importância de adotar soluções tecnológicas que também considerem a sustentabilidade ambiental, reduzindo a necessidade de servidores físicos e promovendo o uso de data centers otimizados em consumo de energia.

Em suma, o desenvolvimento deste sistema não apenas aprimorou as operações da Barbini Construções, mas também consolidou o aprendizado dos integrantes do projeto em computação em nuvem, segurança e design de sistemas. O PI trouxe uma visão completa dos desafios e das soluções aplicáveis a um sistema de gestão em nuvem, representando um passo significativo no crescimento tecnológico e estratégico da empresa.

## REFERÊNCIAS

Amazon Web Services. **Amazon EC2 Pricing**(2023). Disponível em: <<https://aws.amazon.com/ec2/pricing/>>. Acesso em 07/11/2024.

Amazon Web Services. **AWS Solutions and Cloud Tools for Modern Applications** (2023). Disponível em: <<https://aws.amazon.com/>> Acesso em 06/11/2024.

Armbrust, M. **Cloud Computing e Colaboração Empresarial: Desafios e Oportunidades no Brasil**. Rio de Janeiro: Editora Inovação(2021).

Booch, G. **Object-Oriented Analysis and Design with Applications**. Addison-Wesley Professional(2007).

Brock, J. **Cloud Computing for Small and Medium-Sized Enterprises**. Springer(2022).

Buyya, R., Vecchiola, C., & Selvi, S. T. **Mastering Cloud Computing: Foundations and Applications Programming**. Morgan Kaufmann(2013).

Chen, P. P. "The Entity-Relationship Model—Toward a Unified View of Data". ACM Transactions on Database Systems(1976).

Clements, P., & Northrop, L. **Software Product Lines: Practices and Patterns**. Addison-Wesley Professional(2001).

Costa, C. A. **Migração e Gestão de Infraestrutura em Nuvem: Estratégias para Empresas Brasileiras**. São Paulo: Editora Tecnológica(2020).

Cormen, T. H., Leiserson, C., E., Rivest, R. L., & Stein, C. **Introduction to Algorithms**. MIT Press(2009).

Date, C. J. **An Introduction to Database Systems (8ª ed.)**. Addison-Wesley(2012).

Elmasri, R., & Navathe, S. **Fundamentals of Database Systems**. Pearson(2010).

Elmasri, R., & Navathe, S. B. **Fundamentals of Database Systems**. Pearson Education(2015).

Gupta, B., & Gupta, R. **Practical API Security: Securing APIs for Serverless and Microservice Architectures**(2017).

Heuser, C. A. **Projeto de Banco de Dados**. Bookman Editora(2009).

Krug, S. **Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability**. New Riders(2014).

Martin, R. C. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall(2008).

Mell, P., & Grance, T. **The NIST Definition of Cloud Computing**. NIST Special Publication(2011).

Miceli, André. **Empreendedorismo Digital: Inovação e a Transformação de Negócios**. Editora Gente(2020).

Murphy, J. **Cloud Security and Compliance for AWS**. Packt Publishing(2021).

Murugesan, S. "**Harnessing Green IT: Principles and Practices**". IT Professional, 10(1), 24-33(2008).

Richardson, L., & Ruby, S. **RESTful Web Services**. O'Reilly Media(2007).

Silberschatz, A., Korth, H. F., & Sudarshan, S. **Database System Concepts**. McGraw-Hill(2010).

Tilkov, S., & Vinoski, S. "**Node.js: The Future of Web Applications**". IEEE Internet Computing(2010).

Weiss, M. A. **Data Structures and Algorithm Analysis in Java**. Pearson Education(2012).